Iteration-related Various Learning Particle Swarm Optimization for Quay Crane Scheduling Problem

Mingzhu Yu[1], Xuwen Cong[1], Ben Niu[2*], Rong Qu[3]

[1] Department of Transportation Engineering, College of Civil Engineering, Shenzhen University, Shenzhen, 518060, China

[2] College of Management, Shenzhen University, Shenzhen, 518060, China

[3] School of Computer Science, University of Nottingham, Nottingham, NG8 1BB, UK

Abstract. Quay crane scheduling is critical in reducing operation costs at container terminals. Designing a schedule to handling containers in an efficient order can be difficult. For this problem which is proved NP-hard, heuristic algorithms are effective to obtain preferable solutions within limited computational time. When solving discrete optimization problems, particles are very susceptible to local optimum in Standard Particle Swarm Optimization (SPSO). To overcome this shortage, this paper proposes an iteration-related various learning particle swarm optimization (IVLPSO). This algorithm employs effective mechanisms devised to obtain satisfactory quay crane operating schedule efficiently. Superior solutions can save up to 5 hours for handling a batch of containers, thus significantly reduces costs for terminals. Numerical studies show that the proposed algorithm outperforms state-of-the-art existing algorithms. A series of experimental results demonstrate that IVLPSO performs quite well on obtaining satisfactory Pareto set with quick convergence.

Keywords: Improved particle swarm optimization, Quay crane scheduling problem, Various learning mechanism

1 Introduction

Quay crane scheduling is a crucial operation in container terminal management. When a vessel arrives at the container terminal, quay cranes are assigned to unload containers from it. The quay crane scheduling problem consists of quay cranes, containers and bays. Since a vessel contains several bays, to assign which quay crane to which bay in an order can be a complicated problem. The quay crane scheduling problem has been studied by scholars for years. One of the widely studied methods in quay crane scheduling include genetic algorithm (GA) [1] or modified GA. Some

[*] Corresponding author. B. Niu, drniuben@gmail.com

researches add random elements in crossover and mutation [2], or utilize biased mechanism [3] and modified GA combined with simulation [4].

Particle swarm optimization (PSO) was proposed by Kennedy [5] and Eberhart [6], and has been utilized in bunches of researches over many subjects including the quay crane scheduling problems. PSO is a heuristic algorithm simulating a population of birds foraging to food source continuously. Each individual has a foraging position and updates its flight direction based on individual experience and group experience. To overcome the issue that PSO may be trapped into local optimal solution when addressing large-scale problems, scholars introduced mechanisms into improved versions of PSO. These include dynamic adaptive parameters like hierarchical swarm [7, 8]; inertia weight [9] and learning coefficients [10], learning from leaders [8, 11] or other individuals [7, 12] in a hierarchical system to update particles' velocity and position. To balance the exploitation and exploration of PSO, some scholars introduced un-certain elements to increase the diversity of the population by learning from random individuals [8].

In this paper, to address the issue of particles stuck into the local optimum and to accelerate convergence, an iterations-related variable learning particle swarm optimization (IVLPSO) is proposed, integrating new mechanisms learning from the center position and other near-optimal individuals. The idea of center position was proposed by Niu [7], and showed to accelerate convergence. Dynamic inertia weight is also introduced in IVLPSO. Some opposition solutions of high quality solutions (proposed by Ghasemi [12]) are also taken into consideration, (i.e., the number of a dimension is 2, and its opposition solution is -2in [12], it will be adapted in the proposed algorithm). So as to increase the diversity of IVLPSO. All mechanisms we use aim to explore feasible and better solutions in a short time. Compared with standard particle swarm optimization in addressing quay crane scheduling problems [13], IVLPSO shows to perform better than PSO and GA.

This paper is organized as follows: Section 2 presents the mathematical formulation of the quay crane scheduling problem, and Section 3 describes the process of the IVLPSO algorithm. Experimental results are presented in Section 4. Section 5 concludes the paper.

2 Quay Crane Scheduling Problem Formulation

Li [13] proposed a quay crane scheduling problem with the objective of unloading containers from vessel bays with unequal amount of containers as soon as possible, namely, minimizing the total working time. It was assumed that the safety distance between two quay cranes should be no less than two-vessel-bay length. The problem model for the quay crane scheduling problem and the variables are presented in Table 1. The objective function and constrains are presented as follows.

Table 1. Variables and definitions of Quay Crane Scheduling model

Varibles	Definitions
Q	The total amount of quay cranes
B	The total amount of vessel bays of one vessel
T	The time when the last container is unloaded from the vessel
b	The index of bays, $b \in B$
q	The index of quay cranes, $q \in Q$
t	The index of time, $t \in T$
S_b	The time when the container of vessel bay b starts to be unloaded
F_b	The time when all containers of vessel bay b have been unloaded
X_{qbt}	X_{qbt} =1:quay crane q is working at vessel bay b at time t
	X_{qbt} =0:otherwise
P_{qbt}	P_{qbt} =1 :quay crane q stays on vessel bay b at time t
	P_{qbt} =0 :otherwise

Minimize: T

Subject to:

$$F_b - S_b \geq 0, \forall 1 \leq b \leq B \tag{1}$$

$$\sum_{q=1}^{Q} P_{qbt} \leq 1, \forall 1 \leq b \leq B, \forall 1 \leq t \leq T \tag{2}$$

$$\sum_{q=1}^{Q} X_{qbt} \leq 1, \forall 1 \leq b \leq B, \forall 1 \leq t \leq T \tag{3}$$

$$\sum_{b=1}^{B} P_{qbt} = 1, \forall 1 \leq q \leq Q, \forall 1 \leq t \leq T \tag{4}$$

$$\sum_{b=1}^{B} X_{qbt} \leq 1, \forall 1 \leq q \leq Q, \forall 1 \leq t \leq T \tag{5}$$

$$\sum_{b=1}^{B} q' \cdot P_{qbt} - \sum_{b=1}^{B} q \cdot P_{qbt} > 2, \forall 1 \leq q < q' \leq Q, \forall 1 \leq t \leq T \tag{6}$$

$$T = \max_{b} F_b, \forall 1 \leq \mathrm{b} \leq B \tag{7}$$

$$F_b = \max_{t,q} tX_{qbt}, \forall 1 \leq b \leq B \tag{8}$$

$$X_{qbt} = \begin{cases} 1 \\ 0 \end{cases} \forall 1 \leq q \leq Q, \forall 1 \leq b \leq B, 1 \leq t \leq T \tag{9}$$

$$P_{qbt} = \begin{cases} 1 \\ 0 \end{cases} \forall 1 \leq q \leq Q, \forall 1 \leq b \leq B, 1 \leq t \leq T \tag{10}$$

The objective is to minimize the total working time *T*. Constraints (1) set the start working time and finish working time of a vessel bay. Constraints (2) represent that there is no more than one quay crane staying in a bay at any time. Constraints (3) restrict no more than one quay crane working in a bay at any time. Constraints (4) illustrate that each quay crane must stay at one bay in any time. Constraints (5) mean that every quay crane can work for only one bay at any time. Constraints (6) ensure quay cranes will not cross each other and keep two bays length safety distance. Constraints (8) show the relation vessel between a decision variable and finish working time of a vessel bay. Constraints (9) and (10) restrict the domain of the decision variables.

3 IVLPSO for The Quay Crane Scheduling Problem

3.1 Iteration-related Various Learning Particle Swarm Optimization

The standard PSO simulates bird swarm foraging phenomenon, where each particle acts as a bird to update its velocity and position by its own experience and interaction with the others with parameters called inertia weight and learning coefficients. For more information, please refer to [5]. As one of the classical meta-heuristic algorithms, PSO has been applied widely in optimization problems. However, it suffers from being trapped to local Pareto fronts when tackling complex or large-scale problems. To overcome these drawbacks, IVLPSO is proposed in this paper. It is inspired by the diversified learning strategy used in [7] and [12]. The encoding, operations, and algorithm procedures of the proposed IVLPSO are described as follows.

3.1.1 Encoding

Each particle of IVLPSO is regarded as a potential feasible solution for the quay crane scheduling problem. The dimensions of a particle present the unique vessel bay numbers. The dimensions in a particle are integers presenting the working order of the quay crane, and each quay crane is available only if it has accomplished the previous unloading task. Some constraints of the quay crane scheduling problem can be met using this encoding; (A quay crane can only correspond to one vessel bay at any time, and ensures that each vessel bay's mission is completed, showing a job order). Other constraints will be met through a series of operations based on the positions of the quay crane and the target bay. These operations ensure that each particle is satisfying all constraints. An example of the encoding for a particle is illustrated in Fig.1, where the working sequence of the quay crane is from vessel bay 8, 4, 12 … to 15.

8	4	12	3	19	17	1	5	14	11	10	20	9	16	7	13	18	2	6	15

Fig.1. An example of the encoding for a particle

3.1.2 Three Key Operations of IVLPSO

In order to improve the standard PSO to avoid being trapped into local optimum, additional operations are introduced in IVLPSO, described as follows.

(1) Dynamic Inertia Weight

Empirically, value of inertia weight w is set to be always in the range of [0.4, 0.9]. When w is approaching the minimum, PSO has a good performance in exploitation; when w is closer to the maximum, PSO does well in exploration. According to the characteristics of PSO, which is well known on quick convergence, w is set to decrease along with the iterations in IVLPSO. The probability of getting a better solution decreases a lot when the number of iterations exceeds 100, and the dynamic w is thus formulated as in Eq. (11):

$$w = 0.2 + 0.5 \cdot \frac{4 \cdot e^{10} \cdot m}{5M(e^{10} + 1)} \tag{11}$$

where M is the maximum iteration number, m is the present iteration and the value of w is set to between 0.2 and 0.7. The range of M is usually between 100 and 1000, in order to reduce the excessive influence of iterations on the gradient of inertia weight change, e^{a} in the denominator should be much larger than $5M$. e^{10} can be regarded as far greater than $5M$.

(2) Various Learning Mechanisms

Three mechanisms in Eq. (13), Eq. (14) and Eq. (15) are devised as learning exemplar for each particle. These mechanisms broaden the scope of learning and increase the diversity of the updated particles. The change of particle convergence is automatically judged at the late iteration to determine which mechanism is used to update the particles. If the value of the fitness function remains unchanged for dozens of consecutive times, the particle will be updated with Eq. (15), for other ways see the following equations (Eq. (13) and Eq. (14)). The center position of a swarm could lead particles to towards better solutions efficiently. Particles select exemplar to update themselves within the global best position ($gbest$) and center position randomly. Exemplar of each particle is decided by generating a $PC(i)$ in Eq. (12), which was proposed by Niu et al [7], and the velocity of particles is update using Eq. (13).

$$PC(\mathrm{i}) = 0.05 + 0.45 \cdot \frac{e^{\frac{10(i-1)}{M-1}}}{e^{10} - 1} \tag{12}$$

If $R(i) < PC(i)$:

$$v(i+1) = w \cdot v(i) + c_1 \cdot rand \cdot (P_c - x(i)) + c_2 \cdot rand \cdot (P_g - x(i))$$

If $R(i) \geq PC(i)$: (13)

$$v(i+1) = w \cdot v(i) + c_1 \cdot rand \cdot (P_b - x(i)) + c_2 \cdot rand \cdot (P_g - x(i))$$

where i is the index of the particle, $R(\mathrm{i})$ is a random decimal in the range $[0,1]$; $PC(i)$ is the learning probability of particle i. $\mathrm{v}(i)$ and $x(i)$ are the velocity and position of particle i, respectively. P_c is the center position of the swarm and P_b is $pbest$, and P_g is $gbest$. If $R(i)$ is greater than $PC(i)$, particle i learns from $gbest$ and its personal best ($pbest$); otherwise, particle i regards the center position and $gbest$ as its exemplar.

It is also observed that $gbest$ convergences quite quickly in standard PSO, and the curve of solution quality becomes smooth and steady when iteration number reaches about one fifth of the maximum iteration number. The rest of iterations thus could benefit from some randomness in finding better solutions. Two new strategies are proposed in IVLPSO to increase the diversity of the swarm. First, as long as the iteration number reaches one fifth of the maximum iteration number, exemplars become some near-optimal particles (e.g. the half of the total individuals ranked the top in terms of fitness). This strategy is formulated as below in (14).

If iteration > $M/5$ and $R(i) < PC(i)$:

$$v(i+1) = w \cdot v(i) + c_1 \cdot rand \cdot (P_\mathrm{r} - x(i)) + c_2 \cdot rand \cdot (P_g - x(i))$$

If $R(i) \geq PC(i)$: (14)

$$v(i+1) = w \cdot v(i) + c_1 \cdot rand \cdot (P_b - x(i)) + c_2 \cdot rand \cdot (P_g - x(i))$$

where P_r is a random individual among near-optimal individuals of the swarm.

When the iteration number reaches one fifth of the total iterations, $R(\mathrm{i})$ is less than $PC(i)$ and $gbest$ is a constant over many iterations, then the new exemplar is opposition solution (take the opposite value of each number in every dimension. Sometimes the opposite value can be negative in some papers, it equals the maximum minus the current value in this paper). The new exemplar among the newest $pbest$ and three more particles according to the fitness of their opposition solutions, see Eq. (15).

$$P_f = \max\left\{ f(opbest), f(op_1), f(op_2), f(op_3) \right\}$$
$$\mathrm{x}(i) = P_f \tag{15}$$

where P_f is the new exemplar, $f(x)$ is the fitness of x; $opbest$ is the opposition solution of $pbest$; op_1 op_2 op_3 are the opposition solutions of the best three solutions.

(3) Boundary Restrictions

To make sure that solutions satisfy constraints defined in Section 2, and solutions on the boundary of each dimension (each dimension represents a number of vessel bay, so the lower boundary is 1 and the upper boundary is the number of total vessel bays. i.e., 20 vessel bays in total in [13], the upper bound is 20; 30 vessel bays in total, the upper bound is 30…) are not ignored, every dimension should take values within a lower bound and an upper bound after its velocity is updated. Boundary restrictions are made using Eq. (16).

If $x_i > x_{\max}$:

$$x_i = x_{\max}$$

If $x_i < x_{\min}$: (16)

$$x_i = x_{\min}$$

where x_i is the position of the i th dimension. $x_{\max}$ is the maximum of position number, which equals the amount of vessel bays. $x_{\min}$ is the minimum of vessel bay number, which equals to 1.

Because every number in each dimension represents a number of a vessel bay, it can only be a positive integer. But it may become a decimal number after the update. So numbers of all dimensions are ranked and we get a new array that can restore to a large extent its integral parts (otherwise the update doesn't make sense); this operation aims to avoid repetition or non-integer after updating. The original corresponding sequencing number of the array needs to be recorded to obtain a similar solution with no redundancy.

3.2 The IVLPSO Algorithm for Quay Crane scheduling model

Based on mechanism of IVLPSO for terminal quay crane scheduling, the computational experiments are conducted in MATLAB environment. SPSO and GA are chosen as the comparing algorithms by using the same parameters and settings as IVLPSO. (Population size is 30 and iteration is 500). The pseudo-code of IVLPSO in solving quay crane scheduling problem is shown in Table 3.

Table 3. The pseudo-code of IVLPSO

```
Begin
 Initialize parameters and produce the initial population of IVLPSO
 (c1 = c2 = 0.4, M = 500, D = 20, N = 30, vmin = −2, vmax = 2)
For ( i = 1 : N ) // N: no. of iterations
    Calculate the fitness of each particle according to model in Section 3.1;
    Reserve pbest and gbest ;
End
For ( t = 1 : M ) // M: population of particles
   For ( j = 1 : D ) //D: dimensions of the particle
        Update the velocity of every dimension of particles using Eq. (13) and Eq. (14);
        Update position using Eq. (13) and Eq. (15);
        Do boundary restrictions by rules in Eq. (16);
   End
  Update fitness and reserve gbest ;
End
Output: value of gbest (shortest working time of quay cranes) and the solution
End
```

4 Experimental Results and Analysis

4.1 Experiment parameter settings

The parameter settings of the quay crane scheduling problem are the same as the PSO in [13], and the same scale of the problem, which include the amount of cargo of each vessel bay, the working efficiency of quay cranes and their travel time for the problem. More information and related problem data can be referred to [13].

In the instance of [13] (denoted as task(1)), there are in total 20 bays in the arriving vessel. The number of containers that require to be unloaded from each bay is as follows:

$$task(1) = (80,168,180,66,200,180,220,60,50,140,46,210,20,90,160,110,250,50,200,160)$$

To show the efficiency of the proposed IVLPSO algorithm, other instances (task(2) - task(4)) have also been generated with different scale of bays and different number of containers. The total number of bays in these four instances varies from 20 to 40, and the average number of containers in the bays varies from 88 to 195.

$$task(2) = \left(275,47,264,266,282,140,144,250,197,38,207,232,192,73,243, 89,282,280,121,278\right)$$

$$task(3) = \begin{pmatrix} 89,108,69,120,79,123,118,170,183,83,183,99,109,145, \\ 10,13, 81,58,35,8,49,120,21,139,48,125,61,49,44,101 \end{pmatrix}$$

$$task(4) = \begin{pmatrix} 265,173,189,75,139, 99,243,106, 88,177,69,240,24,37,47, \\ 198,39,27,42,145,62,204,269,101,260,257,33,144,8,140 \end{pmatrix}$$

$$task(5) = \begin{pmatrix} 6,223,130,72,199,220,63,42,294,56,68,159,26,97,149,114,171,116,120,17, \\ 255,30,182,34,108,279,236,103,8,66,58,229,33,116,281,245,158,191,37,209 \end{pmatrix}$$

Working efficiency (containers per hour) of three quay cranes is normally distributed with $N(34.6, 2.69)$.

Tables show the results of the five instances with different combination of bays and containers using PSO, GA and IVLPSO on the same computer and same parameters (population size is 30 in three algorithms, learning factors are 0.4 and 0.4 in PSO and IVLPSO and cross probability is 0.8 and mutation probability is 0.2 in GA) , respectively. Instance “20-2640” means the instance considers 20 bays and 2640 containers. These characteristics are set considering the operating data of Shekou terminal in Shenzhen of China in recent years. As shown in Table 4, Table 5, Table 6, Table 7 and Table 8. For each instance the algorithms are run for 10 times to record the computational time, Pareto set of quay cranes total working time of each run, and also the average fitness. To make a fair comparison, the three algorithms are run with the same number of populations (i.e. 30) and same number of iterations (i.e. 500).

As shown in Table 4, Table 6 and Table 8, IVLPSO produces better solutions in a shorter time than PSO and GA. Table 9 shows the solution of the best fitness 93124 for instance 1. Fig. 2 presents the convergence of the three algorithms. Conclusions can be drew as follows:

- IVLPSO performs the best among the three algorithms compared. More than half of the results from IVLPSO are better than the minimum of the other two algorithms. This demonstrates the effectiveness of IVLPSO.
  IVLPSO has showed superiority in quicker convergence without crossover or mutation in GA. Moreover, its diversity is no worse than GA. The convergence of PSO and IVLPSO are similar, except that the latter has a greater decrease in the first 50 iterations. The best solution can save about 2.7 hours per vessel, which lead to potentially huge cost saving for port operators.
- In summary, IVLPSO provides a new approach for port operators to solve the quay crane scheduling problem in an effective and efficient way.

Table 4. The experimental results of the quay crane working time (seconds) for Instance 1.

Instance	Number	PSO	GA	IVLPSO
20-2640	1	103877	95163	**94200**
	2	102567	97893	**93498**
	3	98992	96427	**94476**
	4	102512	97925	95633
	5	104874	97664	97553
	6	97265	97571	93421
	7	103681	95163	95898
	8	107802	97893	97102
	9	100335	95573	**94614**
	10	98354	98268	**93265**
	Avg.	102025	96954	**94966**
Avg. Running Time		2394	4396	**2339**

Table 5. The experimental results of the quay crane working time (seconds) for Instance 2.

Instance	Number	PSO	GA	IVLPSO
20-3900	1	143605	138839	141638
	2	137475	138893	139439
	3	139491	137139	**134333**
	4	148816	140555	**135589**
	5	152829	140004	148456
	6	142762	144090	143680
	7	148462	140499	139001
	8	149269	140597	141145
	9	144116	139643	146840
	10	153120	141353	**136568**
	Avg.	145994	140161	140668
Avg. Running Time		3816	5149	**2577**

Table 6. The experimental results of the quay crane working time (seconds) for Instance 3.

Instance	Number	PSO	GA	IVLPSO
30-2640	1	110064	100829	102385
	2	102508	102712	**95920**
	3	109502	102465	**98089**
	4	108191	101685	**97531**
	5	106970	105857	**99625**
	6	105402	103365	**99662**
	7	111828	103844	**99449**
	8	103092	102272	**96967**
	9	107597	102920	105763
	10	101039	103198	**100124**
	Avg.	106619	102914	**99551**
Avg. Running Time		3401	3800	**3349**

Table 7. The experimental results of the quay crane working time (seconds) for Instance 4.

Instance	Number	PSO	GA	IVLPSO
30-3900	1	148383	144977	142877
	2	157627	142357	150699
	3	152620	141767	148995
	4	163540	145282	**140179**
	5	143826	143005	149207
	6	158534	144288	147549
	7	154602	143572	142624
	8	155703	144393	145696
	9	159499	145712	149781
	10	153832	144245	145907
	Avg.	154816	143959	146351
Avg. Running Time		4906	4762	**3819**

Table 8. The experimental results of the quay crane working time (seconds) for Instance 5.

Instance	Number	PSO	GA	IVLPSO
40-5200	1	209790	209011	**199736**
	2	214227	201734	210672
	3	216649	208632	**198676**
	4	216565	205376	**192594**
	5	218036	200641	205711
	6	210105	204521	204584
	7	212526	212303	**195200**
	8	205123	207575	**196767**
	9	214251	209439	203684
	10	221760	210282	206894
	Avg.	213903	206951	**201452**
Avg. Running Time		4802	7281	5972

Notes: Instance "20-2640" means the task considers 20 bays and 2640 containers.

Table 9. Working schedule of the best solutions for Instance 1.

Bay Num.	17	5	10	3	7	11	14	4	20	18
Crane Num.	3	1	2	1	2	3	3	1	3	2
Task	80	168	180	66	200	180	220	60	50	140

Bay Num.	1	12	6	16	19	13	9	15	2	8
Crane Num.	1	2	1	3	3	2	1	2	1	2
Task	46	210	20	90	160	110	250	50	200	160

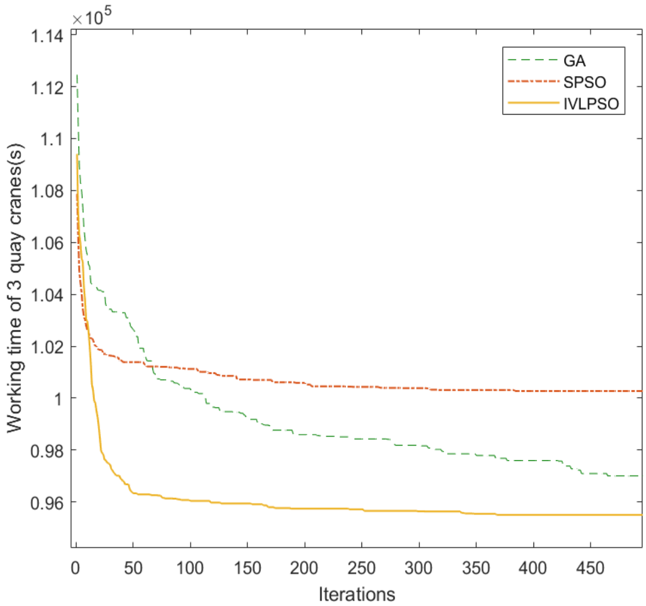

Fig. 2. Convergence of PSO, GA and IVLPSO for Instance 1.

5 Conclusions

This paper proposes an improved PSO (IVLPSO) on the basis of PSO, where various learning mechanisms are employed as the number of iterations changes. The proposed IVLPSO showed to be able to escape from the local optimum, and has a better ability to find better solutions for a quay crane scheduling problem in terms of minimizing the working time. To show its advantages, the proposed new algorithm is compared against GA and PSO over five instances. Experimental results demonstrate the superiority of IVLPSO for the quay crane scheduling problem. IVLPSO converges fast, only requiring one-third of the time that the other two algorithms consume to obtain satisfactory solutions. In our future work, the proposed IVLPSO will be extended to solve multi-objective problems and large-scale problems with more advanced learning mechanisms.

Acknowledgment

This work is partially supported by The National Natural Science Foundation of China (Grants Nos. 71571120, 71271140, 61472257), Natural Science Foundation of Guangdong Province (2016A030310074).

References

1. Chung, S. H., Choy, K. L.: A modified genetic algorithm for quay crane scheduling operations. Expert Systems with Applications, 39(4)：4213-4221(2012).
2. Kaveshgar, N., Huynh, N., Rahimian, S. K.: An efficient genetic algorithm for solving the quay crane scheduling problem. Expert Systems with Applications, 39(18): 13108-13117(2012).
3. Correcher, J. F., Alvarez-Valdes, R.: A biased random-key genetic algorithm for the time-invariant berth allocation and quay crane assignment problem. Expert Systems with Applications, 89(2017).
4. Azevedo, A. T. D., Neto, L. L. D. S., Chaves, A. A., Moretti, A. C.: Solving the 3d stowage planning problem integrated with the quay crane scheduling problem by representation by rules and genetic algorithm. Applied Soft Computing, 65(2018).
5. Kennedy, J., Eberhart, R.: Particle swarm optimization. IEEE International Conference on Neural Networks, Proceedings, vol.4, pp. 1942-1948 (1995).
6. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. International Symposium on MICRO Machine and Human Science . IEEE, pp. 39- 43(2002).
7. Niu, B., Huang, H., Tan, L., Duan, Q.: Symbiosis-based alternative learning multi-swarm particle swarm optimization. IEEE/ACM Trans Comput Biol Bioinform, 14(1), pp. 4-14(2017).
8. Ge, H., Sun, L., Tan, G., Chen, Z., Chen, C. L.: Cooperative hierarchical pso with two stage variable interaction reconstruction for large scale optimization. IEEE Transactions on Cybernetics, 47(9), pp. 2809-2823(2017).
9. Li-Xin Wei, Xin Li, Rui Fan, Hao Sun, Zi-Yu Hu.: A hybrid multi-objective particle swarm optimization algorithm based on r2 indicator. Information Technology, pp(99): 1-1(2018).
10. Tehsin, S., Rehman, S., Saeed, M. O. B., Riaz, F., Hassan, A., Abbas, M., et al.: Self-organizing hierarchical particle swarm optimization of correlation filters for object recognition. IEEE Access, pp(99): 1-1(2017).
11. Zhu, Q., Lin, Q., Chen, W., Wong, K. C., Coello Coello, C. A., Li, J., et al.: An external archive-guided multiobjective particle swarm optimization algorithm. IEEE Transactions on Cybernetics, 47(9), pp. 2794-2808(2017).
12. Kang, Q., Xiong, C. F., Zhou, M. C., Meng, L. P.: Opposition-based hybrid strategy for particle swarm optimization in noisy environments. IEEE Access, pp(99): 1-1.(2018)
13. Haoyuan, L.: Research on simulation Based Optimization Approaches for Logistic systems in Container Port, pp. 86-95(2013).